

What Is Claimed Is:

1 1. A method for performing a lock-free update to one or more fields
2 in an existing node in a linked list, comprising:
3 receiving a reference to the existing node in the linked list, wherein the
4 existing node contains the one or more fields to be updated;
5 obtaining a new node to be added to the linked list, wherein other
6 processes do not possess references to the new node and therefore cannot initially
7 access the new node;
8 copying a snapshot of the existing node to the new node;
9 updating one or more fields in the new node that correspond to the one or
10 more fields in the existing node; and
11 performing a single atomic operation that modifies a next pointer of the
12 existing node to point to the new node and also marks the next pointer to indicate
13 that the existing node is deleted, whereby the new node becomes part of the linked
14 list and the existing node is deleted in a single atomic operation.

1 2. The method of claim 1, wherein after the existing node has been
2 deleted, the method further comprises splicing the existing node out of the linked
3 list by atomically modifying the next pointer of a node immediately preceding the
4 existing node in the linked list to point to the new node, instead of pointing to the
5 existing node.

1 3. The method of claim 2, wherein if a process that deleted the
2 existing node does not perform the splicing operation, another process, which
3 subsequently detects that the existing node has been deleted, performs the splicing
4 operation.

1 4. The method of claim 1, wherein copying a snapshot of the existing
2 node to the new node involves:
3 copying the contents of the existing node to the new node;
4 examining the next pointer of the existing node to determine if the existing
5 node has been deleted; and
6 if the existing node has been deleted, taking a remedial action.

1 5. The method of claim 4, wherein taking the remedial action
2 involves:
3 following the next pointer of the existing node in an attempt to find an
4 updated version of the existing node; and
5 if an updated version of the existing node is found, copying a snapshot of
6 the updated version of the existing node to the new node.

1 6. The method of claim 1, further comprising deleting a target node
2 from the linked list by:
3 receiving a reference to the target node to be deleted from the linked list;
4 atomically marking a next pointer in the target node to indicate that the
5 target node is deleted; and
6 atomically modifying the next pointer of a node immediately preceding the
7 target node in the linked list to point to a node immediately following the target
8 node in the linked list, instead of pointing to the target node, thereby splicing the
9 target node out of the linked list.

1 7. The method of claim 6, wherein after the target node is spliced out
2 of the linked list, the method further comprises modifying the next pointer of the

1 target node so that the next pointer remains marked but points to a node
2 immediately preceding the target node instead of the node immediately following
3 node the target node in the linked list.

1 8. The method of claim 1, further comprising inserting an additional
2 node into the linked list by:
3 identifying a node immediately preceding the additional node in the linked
4 list;
5 identifying a node immediately following the additional node in the linked
6 list; and
7 splicing the additional node into the linked list by,
8 setting the next pointer for the additional node to point to
9 the immediately following node, and
10 atomically updating the next pointer of the immediately
11 preceding node to point to the additional node.

1 9. The method of claim 1, further comprising reading a snapshot of
2 multiple fields from a target node in the linked list by:
3 reading the multiple fields from the target node;
4 examining the next pointer of the target node to determine if the target
5 node has been deleted; and
6 if the target node has been deleted, taking a remedial action.

1 10. The method of claim 9, wherein taking the remedial action
2 involves:
3 following the next pointer of the target node in an attempt to find an
4 updated version of the target node; and

5 if an updated version of the target node is found, repeating the process of
6 reading a snapshot of the multiple fields from the updated version of the target
7 node.

1 11. The method of claim 1, wherein atomically modifying the next
2 pointer of the existing node to indicate that the existing node is deleted involves
3 setting a “deleted bit” in the next pointer.

1 12. The method of claim 1, wherein while atomically modifying the
2 next pointer of the existing node, if the next pointer indicates that the existing
3 node is already deleted, the atomic modification operation fails and the method
4 further comprises taking a remedial action to deal with the fact that the existing
5 node is already deleted.

1 13. The method of claim 1, wherein a given node in the linked list
2 includes:
3 a key that contains an identifier for the given node;
4 one or more fields containing data values or pointers to data values
5 associated with the given node; and
6 a next pointer that contains the address of a node that immediately follows
7 the given node in the linked list, and that also contains a deleted indicator, which
8 indicates whether the given node has been deleted.

1 14. The method of claim 1, further comprising periodically performing
2 a garbage-collection operation to reclaim deleted nodes that have become
3 unreachable.

1 15. A computer-readable storage medium storing instructions that
2 when executed by a computer cause the computer to perform a method for
3 performing a lock-free update to one or more fields in an existing node in a linked
4 list, the method comprising:
5 receiving a reference to the existing node in the linked list, wherein the
6 existing node contains the one or more fields to be updated;
7 obtaining a new node to be added to the linked list, wherein other
8 processes do not possess references to the new node and therefore cannot initially
9 access the new node;
10 copying a snapshot of the existing node to the new node;
11 updating one or more fields in the new node that correspond to the one or
12 more fields in the existing node; and
13 performing a single atomic operation that modifies a next pointer of the
14 existing node to point to the new node and also marks the next pointer to indicate
15 that the existing node is deleted, whereby the new node becomes part of the linked
16 list and the existing node is deleted in a single atomic operation.

1 16. The computer-readable storage medium of claim 15, wherein after
2 the existing node has been deleted, the method further comprises splicing the
3 existing node out of the linked list by atomically modifying the next pointer of a
4 node immediately preceding the existing node in the linked list to point to the new
5 node, instead of pointing to the existing node.

1 17. The computer-readable storage medium of claim 16, wherein if a
2 process that deleted the existing node does not perform the splicing operation,
3 another process, which subsequently detects that the existing node has been
4 deleted, performs the splicing operation.

1 18. The computer-readable storage medium of claim 15, wherein
2 copying the snapshot of the existing node to the new node involves:
3 copying the contents of the existing node to the new node;
4 examining the next pointer of the existing node to determine if the existing
5 node has been deleted; and
6 if the existing node has been deleted, taking a remedial action.

1 19. The computer-readable storage medium of claim 18, wherein
2 taking the remedial action involves:
3 following the next pointer of the existing node in an attempt to find an
4 updated version of the existing node; and
5 if an updated version of the existing node is found, copying a snapshot of
6 the updated version of the existing node to the new node.

1 20. The computer-readable storage medium of claim 15, wherein the
2 method further comprises deleting a target node from the linked list by:
3 receiving a reference to the target node to be deleted from the linked list;
4 atomically marking a next pointer in the target node to indicate that the
5 target node is deleted; and
6 atomically modifying the next pointer of a node immediately preceding the
7 target node in the linked list to point to a node immediately following the target
8 node in the linked list, instead of pointing to the target node, thereby splicing the
9 target node out of the linked list.

1 21. The computer-readable storage medium of claim 20, wherein after
2 the target node is spliced out of the linked list, the method further comprises

1 modifying the next pointer of the target node so that the next pointer remains
2 marked but points to a node immediately preceding the target node instead of the
3 node immediately following node the target node in the linked list.

1 22. The computer-readable storage medium of claim 15, wherein the
2 method further comprises inserting an additional node into the linked list by:
3 identifying a node immediately preceding the additional node in the linked
4 list;
5 identifying a node immediately following the additional node in the linked
6 list; and
7 splicing the additional node into the linked list by,
8 setting the next pointer for the additional node to point to
9 the immediately following node, and
10 atomically updating the next pointer of the immediately
11 preceding node to point to the additional node.

1 23. The computer-readable storage medium of claim 15, wherein the
2 method further comprises reading a snapshot of multiple fields from a target node
3 in the linked list by:
4 reading the multiple fields from the target node;
5 examining the next pointer of the target node to determine if the target
6 node has been deleted; and
7 if the target node has been deleted, taking a remedial action.

1 24. The computer-readable storage medium of claim 23, wherein
2 taking the remedial action involves:

3 following the next pointer of the target node in an attempt to find an
4 updated version of the target node; and
5 if an updated version of the target node is found, repeating the process of
6 reading a snapshot of the multiple fields from the updated version of the target
7 node.

1 25. The computer-readable storage medium of claim 15, wherein
2 atomically modifying the next pointer of the existing node to indicate that the
3 existing node is deleted involves setting a “deleted bit” in the next pointer.

1 26. The computer-readable storage medium of claim 15, wherein while
2 atomically modifying the next pointer of the existing node, if the next pointer
3 indicates that the existing node is already deleted, the atomic modification
4 operation fails and the method further comprises taking a remedial action to deal
5 with the fact that the existing node is already deleted.

1 27. The computer-readable storage medium of claim 15, wherein a
2 given node in the linked list includes:
3 a key that contains an identifier for the given node;
4 one or more fields containing data values or pointers to data values
5 associated with the given node; and
6 a next pointer that contains the address of a node that immediately follows
7 the given node in the linked list, and that also contains a deleted indicator, which
8 indicates whether the given node has been deleted.

1 28. The computer-readable storage medium of claim 15, wherein the
2 method further comprises periodically performing a garbage-collection operation
3 to reclaim deleted nodes that have become unreachable.

1 29. An apparatus that performs a lock-free update to one or more fields
2 in an existing node in a linked list, comprising:
3 a node obtaining mechanism configured to obtain a new node to be added
4 to the linked list, wherein other processes do not possess references to the new
5 node and therefore cannot initially access the new node;
6 a copying mechanism configured to copy a snapshot of the existing node
7 to the new node;
8 an updating mechanism configured to update one or more fields in the new
9 node that correspond to the one or more fields in the existing node; and
10 a modification mechanism configured to perform a single atomic operation
11 that modifies a next pointer of the existing node to point to the new node and also
12 marks the next pointer to indicate that the existing node is deleted, whereby the
13 new node becomes part of the linked list and the existing node is deleted in a
14 single atomic operation.

1 30. The apparatus of claim 29, further comprising a splicing
2 mechanism, wherein after the existing node has been deleted, the splicing
3 mechanism is configured to splice the existing node out of the linked list by
4 atomically modifying the next pointer of a node immediately preceding the
5 existing node in the linked list to point to the new node, instead of pointing to the
6 existing node.

1 31. The apparatus of claim 30, wherein if a process that deleted the
2 existing node does not activate the splicing mechanism, another process, which
3 subsequently detects that the existing node has been deleted, activates the splicing
4 mechanism.

1 32. The apparatus of claim 29, wherein the copying mechanism is
2 configured to:
3 copy the contents of the existing node to the new node;
4 examine the next pointer of the existing node to determine if the existing
5 node has been deleted; and
6 if the existing node has been deleted, to take a remedial action.

1 33. The apparatus of claim 32, wherein while taking the remedial
2 action, the copying mechanism is configured to:
3 follow the next pointer of the existing node in an attempt to find an
4 updated version of the existing node; and
5 if an updated version of the existing node is found, to copy a snapshot of
6 the updated version of the existing node to the new node.

1 34. The apparatus of claim 29, further comprising a deletion
2 mechanism configured to delete a target node from the linked list, wherein the
3 deletion mechanism is configured to:
4 receive a reference to the target node to be deleted from the linked list;
5 atomically mark a next pointer in the target node to indicate that the target
6 node is deleted; and to
7 atomically modify the next pointer of a node immediately preceding the
8 target node in the linked list to point to a node immediately following the target

9 node in the linked list, instead of pointing to the target node, thereby splicing the
10 target node out of the linked list.

1 35. The apparatus of claim 34, wherein after the target node is spliced
2 out of the linked list, the deletion mechanism is configured to modify the next
3 pointer of the target node so that the next pointer remains marked but points to a
4 node immediately preceding the target node instead of the node immediately
5 following node the target node in the linked list.

1 36. The apparatus of claim 29, further comprising an insertion
2 mechanism configured to insert an additional node into the linked list, wherein the
3 insertion mechanism is configured to:
4 identify a node immediately preceding the additional node in the linked
5 list;
6 identify a node immediately following the additional node in the linked
7 list; and to
8 splice the additional node into the linked list by,
9 setting the next pointer for the additional node to point to
10 the immediately following node, and
11 atomically updating the next pointer of the immediately
12 preceding node to point to the additional node.

1 37. The apparatus of claim 29, further comprising a reading
2 mechanism configured to read a snapshot of multiple fields from a target node in
3 the linked list, wherein the reading mechanism is configured to:
4 read the multiple fields from the target node;

5 examine the next pointer of the target node to determine if the target node
6 has been deleted; and
7 if the target node has been deleted, to take a remedial action.

1 38. The apparatus of claim 37, wherein while taking the remedial
2 action, the reading mechanism is configured to:
3 follow the next pointer of the target node in an attempt to find an updated
4 version of the target node; and
5 if an updated version of the target node is found, to repeat the process of
6 reading a snapshot of the multiple fields from the updated version of the target
7 node.

1 39. The apparatus of claim 29, wherein while atomically modifying the
2 next pointer of the existing node to indicate that the existing node is deleted, the
3 modification mechanism is configured to set a “deleted bit” in the next pointer.

1 40. The apparatus of claim 29, wherein while atomically modifying the
2 next pointer of the existing node, if the next pointer indicates that the existing
3 node is already deleted, the modification mechanism is configured to:
4 fail the modification operation fails; and to
5 take a remedial action to deal with the fact that the existing node is already
6 deleted.

1 41. The apparatus of claim 29, wherein a given node in the linked list
2 includes:
3 a key that contains an identifier for the given node;

1 one or more fields containing data values or pointers to data values
2 associated with the given node; and
3 a next pointer that contains the address of a node that immediately follows
4 the given node in the linked list, and that also contains a deleted indicator, which
5 indicates whether the given node has been deleted.

1 42. The apparatus of claim 29, further comprising a garbage collection
2 mechanism configured to periodically perform a garbage-collection operation to
3 reclaim deleted nodes that have become unreachable.